IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 15, 2020, accepted October 24, 2020, date of publication October 28, 2020, date of current version November 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034399

Prototype Cross Platform Oriented on Cybersecurity, Virtual Connectivity, Big Data and Artificial Intelligence Control

ALESSANDRO MASSARO[1], (Senior Member, IEEE), MICHELE GARGARO[1], GIOVANNI DIPIERRO[1], ANGELO MAURIZIO GALIANO[1], AND SIMONE BUONOPANE[2]

[1]Dyrecta Lab, Research Institute, 70014 Conversano, Italy
[2]Spacertron S.r.l., 20131 Milan, Italy

Corresponding author: Alessandro Massaro (alessandro.massaro@dyrecta.com)

ABSTRACT This article describes a prototype cross platform based on intelligent switching of Virtual Private Network (VPN) communications by means of artificial intelligence algorithms able to identify and classify attack risks in self-learning mode by analysing the traffic logs of the system. The platform is also suitable for disaster recovery, data migration and ensures virtualization of communications between nodes in case of risk detection. In order to test the models and evaluate the accuracy of the AI algorithms for risk detection and classification, a number of cyberattack scenario have been simulated. The proposed platform implements Cassandra Big Data system interfacing with supernodes enabling data migration, security and disaster recovery. By comparing the performance of different AI algorithms, the results show that a XGBoost-based algorithm is the most efficient and accurate method for cyberattacks prevention, showing a remarkable ability of classifying and identifying characteristic patterns of the most representative traffic log variables. The research work has been carried out within the framework of a research industry project.

INDEX TERMS Cybersecurity, artificial intelligence, big data, switching virtualization, data security.

I. INTRODUCTION

A research topic gaining popularity in recent years is the study of new cybersecurity prevention techniques and innovative encryption methodologies [1]. A particular communication system to protect a network from cyberattacks is based on the creation of Virtual Private Network (VPN) channels. VPNs can be automatically reconfigured [2], thus suggesting the idea to switch among them under cyberattack conditions through appropriate security procedures in self-configuration mode by means of cryptographic protocols. Specifically, the VPN channels must guarantee confidentiality, entity authentication, data integrity, authentication, secure access control, availability and minimal amount of security relevant configuration. The physical connection of the VPN channels is different from the logical one: in this direction the overlay network level can be distinguished from the physical network level [3]. Firewall and VPN gateway drop-in-router solutions can be embedded into VPN channels [4] allowing the secure management of network elements as routers, hubs and switches using encrypted tunnels. One of the most important aspects concerning the validation of a new cybersecurity model is the evaluation of network performance: the server throughput represents the transmission capacity and could provide important indications about the network loaded by safety elements [5]. Data [6] and database security [7] are the main issue of the industry research, especially when volume, velocity, variety and veracity of data can be satisfied as for big data systems capable of transferring structured and unstructured data [8]. The database threats have been classified in [9]. The platform design idea will therefore consist of a primary pre-classification of potential attacks, which can be traced and isolated also through the use of artificial intelligence. In this context, in [10] it has been analyzed a model to act primarily on the prevention of the attack adopting artificial intelligence (AI) thus raising the security level. AI can therefore act as an expert system capable of assessing different parameters and variables identifying the attacks in advance [11] and enabling network switching [12]. Recent investigations of AI application for intrusion detection, malware analysis, and spam detection [13]–[41] have shown that the cybersecurity by means of AI algorithms is a topic of continuous research interest. Different AI algorithms can be applied for identification and classification of traffic

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

logs. Intrusion detection [46], [47] is typically performed by Convolutional Neural Networks (CNNs) [48]–[53] and deep learning algorithms [54], [55], [65], [66]. Moreover, if the platform is designed to handle Big Data, the storage system could be interfaced by supernodes, which are responsible for the management of the data flow. The use of supernodes [42]–[45] represents a further possibility of strengthening data security, by means of cryptographic techniques and controlling the system at different levels such as organization layer, distribution layer and access layer. A recent survey on the topic [63] found that modern industrial control systems based on big data analytics and cloud computing would become more secure by exploiting the recent advancements of machine learning algorithms.

Following the guideline of recent research in the field, the paper introduces an innovative approach based on the integration of AI methods in an automated system enabling secure VPN switching by predicting attacks, data managing, data cryptography, and data disaster recovery adopting big data systems. By comparing the performance of the most commonly used classification AI algorithms (e.g. tree-based technique and neural networks), the paper is devoted to find the most accurate algorithm to implement on the platform in order to ensure data security against cyberattacks. Different dataset and data flow simulations have been considered in order to simulate and to reply cyberattack phenomena [56]–[60], thus suggesting an approach to adopt for the experimentation.

The paper is organized as follows. In Sect. II the research industry project specifications are introduced, describing the design of the architecture of the cross platform. Sect. III describes the generated dataset of traffic logs and the preprocessing methods to standardize the dataset. In Sect. IV are discussed and tested different AI algorithms for identifying cyberattacks while in Sect. V are described the technique used for data encryption, data migration and disaster recovery. Finally, the platform testing is described in Sect VI while conclusions are drawn in Sect. VII.

II. MAIN PROJECT SPECIFICATIONS AND SYSTEM ARCHITECTURE

The server used to test the AI attack-prevention algorithms implemented in the platform is made up of the elements reported in Table 1 [61].

The prototype multimedia platform used for experimentation is sketched in the system architecture shown in Fig. 1, made up of the following main elements (see Table 1):

- Access Switch Layer2+ 10/100/1000 Mbps / Core Switch Layer2+ 10/100/1000 Mbps: this is a networking hardware connecting each rack on a computer network by using a packet switching to receive and forward data at the network layer 3 of ISO/OSI standard model;
- IPS/IDS/FW/DNS: couple of firewalls installed after the access switch layer that operate in an exogen mode which purpose is to detect and prevent some intrusions or attacks from external clients;

TABLE 1. Elements of the server used for the experimentation.

Element	Function
Access Switch Layer2+ 10/100/1000 Mbps / Core Switch Layer2+ 10/100/1000 Mbps	Hardware connecting racks using packet switching at network of layer 3 of ISO/OSI model
IPS/IDS/FW/DNS	Exogen firewall
Firewall	Endogen firewall
Notification server	Diagnostic messaging
Registration server	Storing of all events detecting anomalies
Syslog Server	Logs generation and logs storage
NAS Server	File-level computer data storage server
VPN Server	VPN connection
SIP Server	Session Initial Protocol (SIS) signaling protocol
Antivirus	Detect and remove malware or any type of malicious item
TMMS/APP	Manage Mobile Devices (MDM) and Mobile Apps (MAM)
Chat Server	Managing chat services
PhoneBook Server	Managing VOIP, chat and mail services
Mail Server	Server managing mail services
System SMS App Sw Server	Server managing SMS services
System NMS App Sw Server	Managing the devices on the network
SRV Front End Web App	Server dedicated for web applications of the system
SRV Core Process	Elaboration and processing of services
SRV Database	Database management

- VPN Server: this component allows the connection of the rack to a computer network through a VPN, improving the security of the system;
- NAS Server: the Network Attached Storage (NAS) Server is a file-level computer data storage server connected to a computer network specialized in serving files either by its hardware, software or configuration;
- Antivirus: this component is a computer program used to prevent, detect and remove malware or any type of malicious item which escaped from the detection of the firewall.

These elements are integrated in the platform in order to ensure maximum security. The proposed hardware is suitable for AI and big data testing [61]. The cross platform integrates different functionalities expressed by the following specifications:

- Intelligent switching of VPN communications: an artificial intelligence engine will perform the intelligent and safe management of the communications of the VPN channels. The channel switching process is controlled by the AI engine constituting the control room able to evaluate the most secure channel. The AI engine decides on the switching modality based on self-learning models capable of assessing potential forms of attack on the

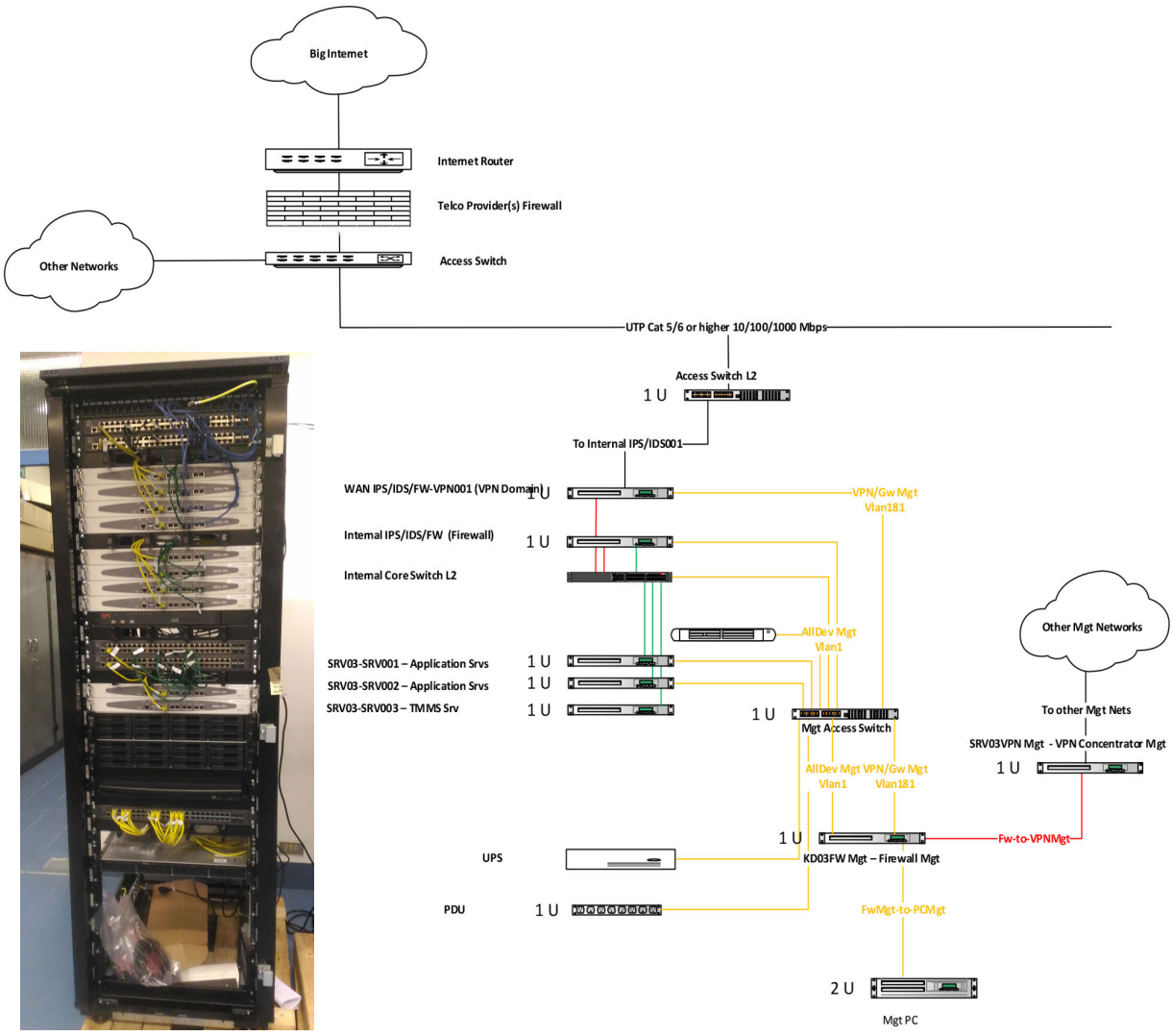

FIGURE 1. Testing hardware used for the experimentation.

network; in addition, the engine maps, the attacks and the network anomalies managing the risk level, the disaster recovery backup by accessing supernodes, data migration between two big data systems proving data security, supernode management and the virtualization of communication between nodes through appropriate association rules;

- Big Data systems: two big data systems enabling the disaster recovery function are integrated into the platform system, following a data migration synchronization by a defined logic (migration from the external databases of each external node to the big data system);
- Virtualization of communications between nodes: the methods of interconnection between the grid connected nodes is designed using appropriate association rules; the dynamic assignment of VPNs and names of individual nodes increases the security level; all transferred data are encrypted thus improving cyber security. The concept of virtual switching comes from the possibility to provide multiple security levels based on a pre-classification of the attacks and on a dynamic threats clustering considering different aspects such as IP analysis, data consistency analysis, linking traceability, tracing the path of the virtualized connection, etc.

In the project four VPN channels were configured over which the intelligent switching engine operates. In order to verify the virtualization features of the network, the connection is traced to detect the path followed by an IP packet to reach a known destination such as the google domain name system (8.8.8.8). Figure 2 shows the network tracing before the VPN configuration, while Fig. 3 shows the data packet tracing starting from one of the four VPN channels configured for the test.

When the network is virtualized (Fig. 3), the path of the packet transits from an additional IP address (192.168.73.1) at the beginning of the path due to virtualization of the

```
PS C:\Users\Utente> tracert 8.8.8.8

 12 ms     1 ms    <1 ms  192.168.70.1
 12 ms     1 ms    <1 ms  192.168.70.1
 77 ms   106 ms   143 ms  172.17.145.254
143 ms   123 ms    88 ms  172.17.145.236
 70 ms    47 ms    38 ms  172.19.245.77
116 ms   124 ms   129 ms  etrunk14.milano50.mil.seabone.net [93.186.128.241]
 67 ms    89 ms   123 ms  72.14.195.206
 66 ms    41 ms    40 ms  108.170.245.81
 38 ms    69 ms    65 ms  216.239.50.241
 69 ms    59 ms    52 ms  dns.google [8.8.8.8]
```

FIGURE 2. Data Packet tracking before the creation of the virtualized network.

```
PS C:\Users\Utente> tracert 8.8.8.8

  1 ms     1 ms     1 ms  192.168.73.1
  2 ms     1 ms     1 ms  192.168.70.1
 86 ms    99 ms   100 ms  172.17.145.254
134 ms   101 ms    97 ms  172.17.145.236
 42 ms    46 ms    41 ms  172.19.245.77
156 ms    99 ms   101 ms  etrunk14.milano50.mil.seabone.net [93.186.128.241]
127 ms   204 ms   117 ms  72.14.195.206
 45 ms    40 ms    52 ms  108.170.245.81
  *      366 ms   204 ms  216.239.50.241
122 ms    37 ms    37 ms  dns.google [8.8.8.8]
```

FIGURE 3. Tracking of the data packet after the creation of the virtualized network.

network. Fig. 4 shows the whole architecture of the digital cross platform described above.

The Unified Modeling Language (UML) diagram shown in Fig. 5 illustrates the main actors of the platform, i.e. the System and the User. The first one carries out all the back-end activities of the prototype system including artificial intelligence data processing and the second one can access to the results shown in a management dashboard which can be referred to as control room.

The executive architecture of the full platform is sketched as a flow chart in Fig. 6. The artificial intelligence algorithm also includes the analysis of the number of attacks carried out in a limited period of time, evaluating if the connection requests are more than the average connection requests. Furthermore, the attack frequencies are saved in the database in order to evaluate a possible correlation between the attack and the time on which it occurs. The network management is performed by the virtualized control room which takes care of both the launch of the log classification script and the actions to be taken in the event of an attack: Backup, disaster recovery and switch of VPN communications. In addition, the backup system is managed by the control room enabling data transfer to big data systems through the function called "Data migration". Finally, the yellow boxes show the tasks related to the hashing of the learning model and the private key for network virtualization. Specifically, in the case of a high number of connections in short time or occuring at the same time in adjacent days, a random switching is carried out ensuring security. The virtualized network has been implemented through *pfsense* (open source software tool for firewall and routing functions).

III. ATTACK LOGS SYSTEM

In order to test the artificial intelligence algorithms a number of traffic logs in different conditions has been generated. The traffic logs are related to the following conditions:

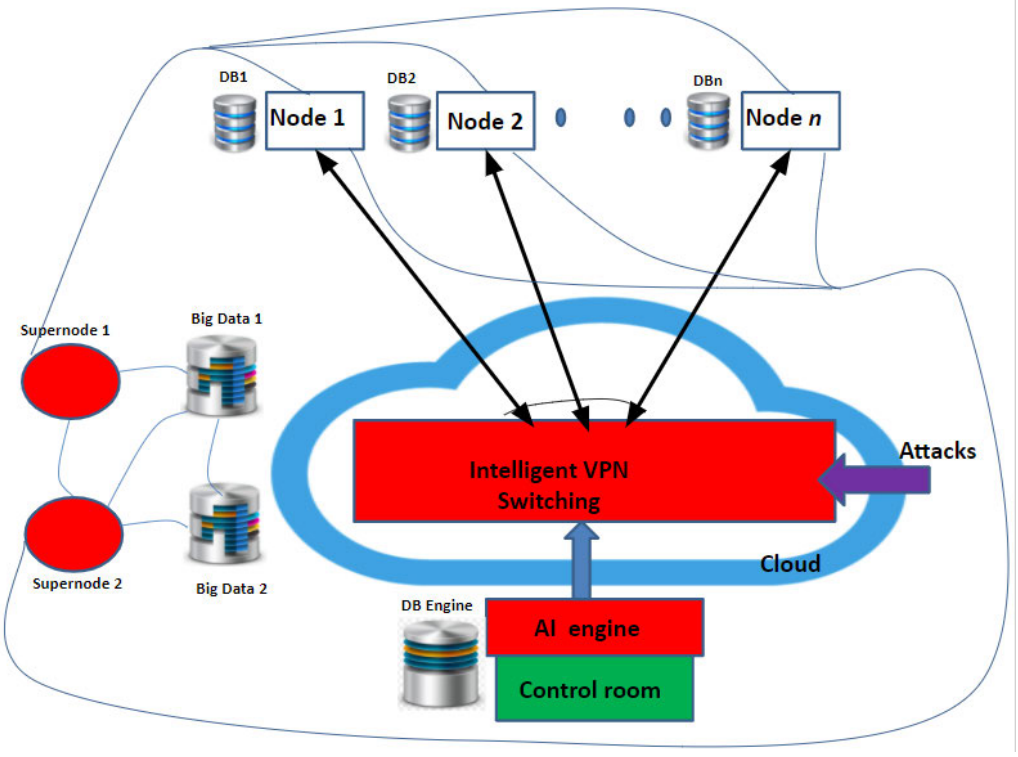

FIGURE 4. Preliminary architecture of the digital cross platform oriented on virtual AI distribution network processes.

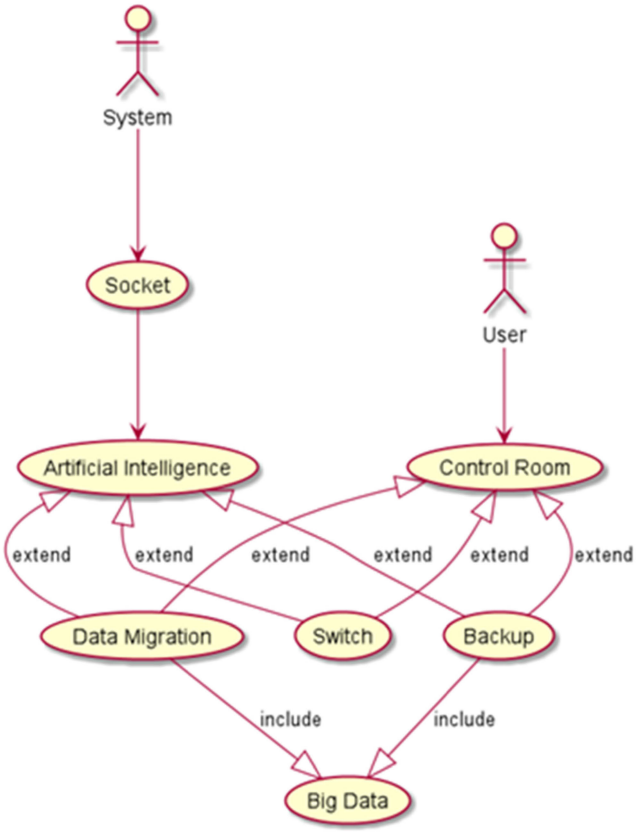

FIGURE 5. UML diagram representing main system actors of the platform.

- Normal traffic: about 40,000 lines of traffic logs in non-attack conditions;
- Denial of Service (DoS) Attack Traffic: 31500 rows of traffic logs during a simulated Dos attack;
- Traffic with BruteForce attack: 500 lines of traffic logs during a simulated BruteForce attack;
- Traffic with Port Scanning attack: 500 lines of traffic logs during a simulated Port Scanning attack.

These attacks represent a set of possible attacks that can occur and allows to fully evaluate the response of the system and the accuracy of the algorithm. The analysis was focused on server-side attacks. Since the artificial intelligence engine is based on the continuous learning of traffic logs updated in real time, the model relies on the correct labelling of the training dataset. In Table 2 and in Fig. 7 are listed the set of features that can be shown in the traffic logs. The artificial intelligence engine is based on the processing of these features with the aim to identify and classify cyberattacks.

The main processed parameters indicated in Table 2 are:

- Tracker: unique ID for the firewall rule applied;
- Interface: interface used by the firewall that activates the log;
- reason_entry: reason why the rule was activated;
- action_taken: action performed by the firewall for the rule;
- direction_traffic: direction of traffic-entry / exit;

TABLE 2. List of logs features selected for classification.

Feature Name	Type	Possible Values
rule_number	int	Generic
subrule_number	int	Generic
anchor	text	Generic
tracker	int	Generic
interface	text	ovpns3|igb1|igb0|em0
reason_entry	text	typically 'match'
action_taken	Boolean	Pass|Block
direction_traffic	Boolean	in|out
ip_version	int	4|6
Type of Service (ToS)	hex	0x0
Time to live (TTL)	int	Generic
id	int	Generic
Offset	int	Generic
Flag	text	DF|MF|none|Reserved
Protocol_ID	int	6|17|1 (More used)
Protocol text	text	tcp|udp|icmp (More used)
Length	int	Generic
Source IP	ip_address	From 0.0.0.0 to 255.255.255.255
Destination IP	ip-address	From 0.0.0.0 to 255.255.255.255
Source Port	int	From 0 to 65535
Destination Port	int	From 0 to 65535
Data Length	int	Generic
TCP flag	char	S|A||F|R|P|U|E|W|none(udp)
Sequence Number	int	Generic
ACK	int	Generic
Window	int	Generic
URG	data	Generic
Options	data	mss;sackOK;TS;nop;wscale
Delta Time	float	Generic
Count_log_5s	int	Generic

- ip_version: communication protocol: IPv4 or IPv6;
- TTL: package validity time;
- Id: package ID:
- Offset: offset of a particular fragment relative to the beginning of the original IP packet (the first fragment has as offset the value 0);
- Flag: control of protocol and fragmentation of datagrams;
- proto_text: IP protocol;
- length: length in bytes of the packet;
- source_IP: source IP from which the logged traffic started;
- source_port: source port number;
- dest_port: destination port number;
- data_length: length of the payload;
- TCP_flag: flag that identifies the action to organize communication and data processing;
- sequence_num: the sequential number indicates the first byte of the attached payload or is sent during the establishment and / or removal of the connection. It serves at

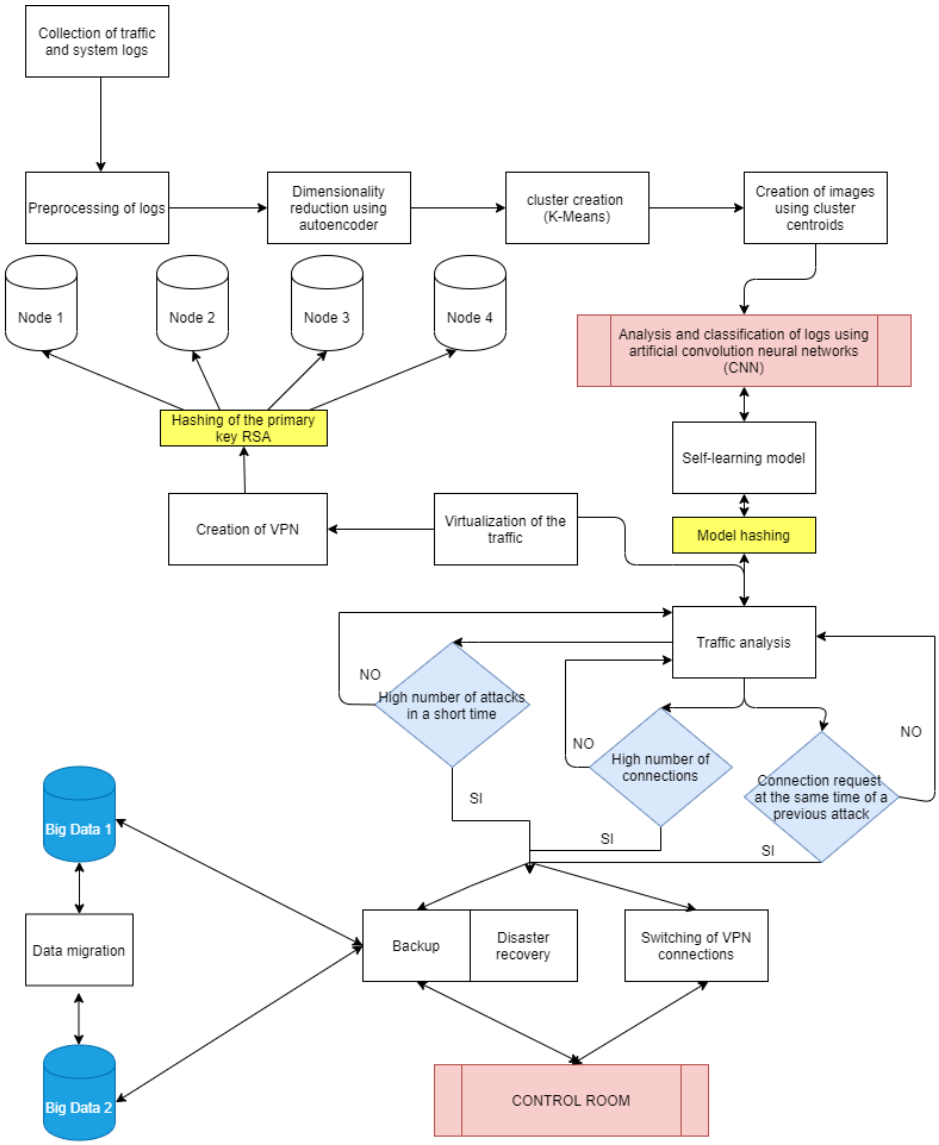

FIGURE 6. Executive architecture of the cybersecurity platform.

16:17:19 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,25399,0,DF,6,tcp,52,192.168.71.2,192.168.0.2,58148,389,0,S,3496339916,,64240,,mss;nop;wscale;nop;nop;sackOK,Normal
16:17:19 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,1442,0,DF,6,tcp,52,192.168.71.2,192.168.1.255,58149,1688,0,S,4093869198,,64240,,mss;nop;wscale;nop;nop;sackOK,Normal
16:17:29 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,11355,0,none,17,udp,107,192.168.71.2,192.168.0.25,54777,161,87,,,,,,,Normal
16:17:34 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,35680,0,none,17,udp,60,192.168.71.2,208.67.222.222,61999,53,40,,,,,,,Normal
16:17:34 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,13404,0,DF,6,tcp,52,192.168.71.2,216.58.205.68,58150,443,0,S,4104912632,,64240,,mss;nop;wscale;nop;nop;sackOK,Normal
16:17:38 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,35682,0,none,17,udp,61,192.168.71.2,208.67.222.222,65374,53,41,,,,,,,Normal
16:17:38 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,42563,0,DF,6,tcp,52,192.168.71.2,216.58.209.35,58151,443,0,S,671060499,,64240,,mss;nop;wscale;nop;nop;sackOK,Normal
16:17:38 filterlog:	78,,,1584436856,ovpns3,match,pass,in,4,0x0,,128,35684,0,none,17,udp,77,192.168.71.2,208.67.222.222,58207,53,57,,,,,,,Normal

FIGURE 7. Example of processed dataset.

the same time to validate and to order (after the transfer) the segments of the data package;

- window: number of bytes that the recipient is able to receive;
- options: requests for activation of TCP functions not included in the general header (Maximum Segment Size-MSS, Window Scaling-WSCALE, Selective Acknowledgments-SACK, No Option-NOP, Timestamps-TS).

Due to the different nature of these features (e.g., categorical, quantitative), the preprocessing phase included the standardization of the features using the widely known techniques *StandardScaler* and *OneHotEncoding* (see Appendix). For example, the categorical feature 'interface' can take the values 'igb1' or 'vpn'. Thus, by applying the one hot encoding the features 'interface' is replaced by two features named 'interface_igb1' and 'interface_vpn' that can take the binary values 0 or 1. In some cases, the presence of some unspecified

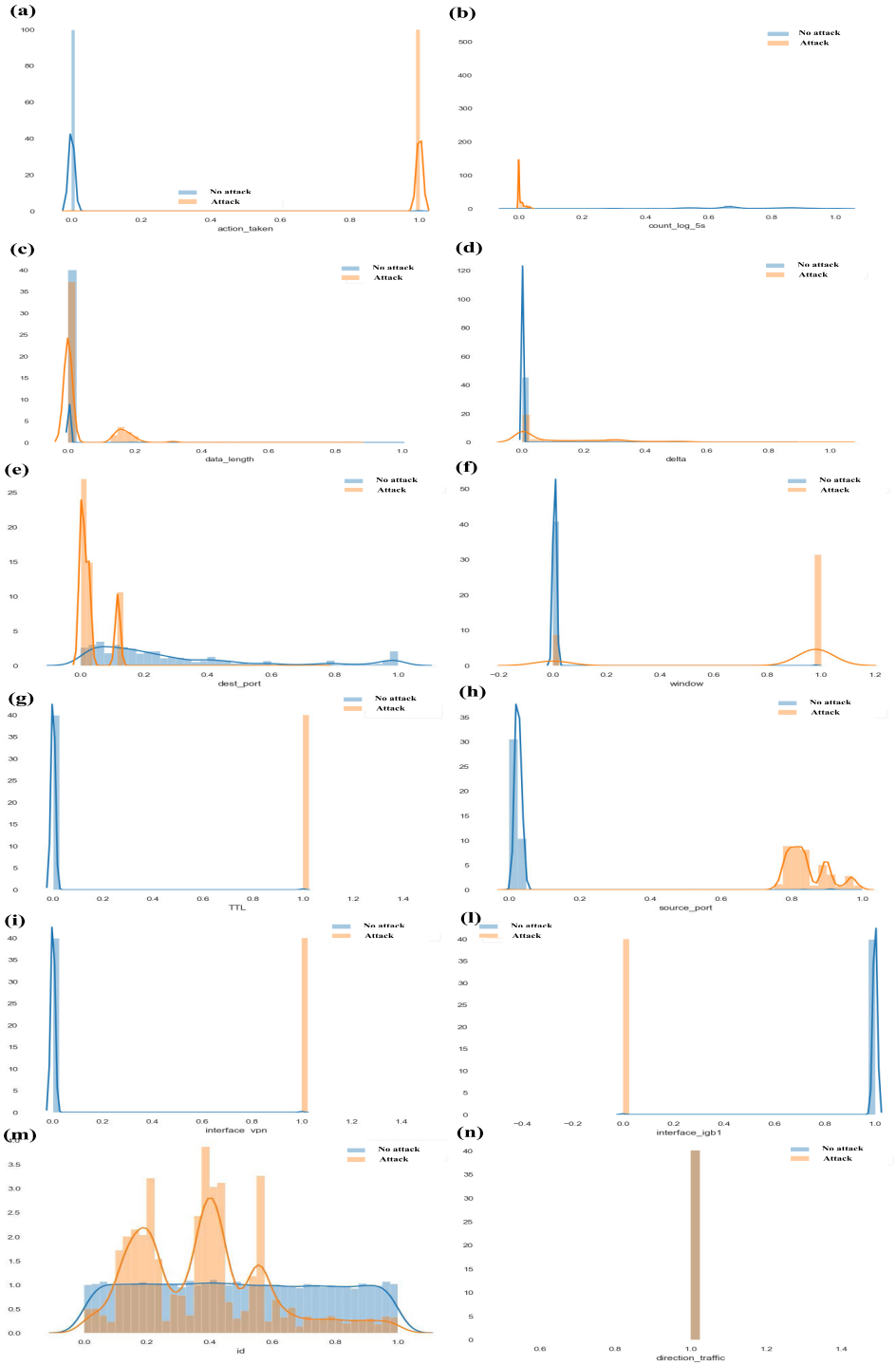

FIGURE 8. Statistical distribution of the log features indicating attacks.

or missing values (such as Not a Number data, NaN) might occurr. In these circumstances we chose to insert the average value of the feature computed on a subset of data showing the same values for the other features.

An initial rough analysis of the traffic logs generated during cyberattacks revealed that the frequency with which transactions are written in the traffic log files is much more related to the presence of the attack than other features. In fact, when the attack starts, the log file is updated much more frequently. Moreover, in order to improve the cyberattacks identification accuracy, two new features were computed by processing the traffic logs:

- Delta Time: the time elapsed between a log and its next;
- Number of logs shown in a specific time interval (10 seconds).

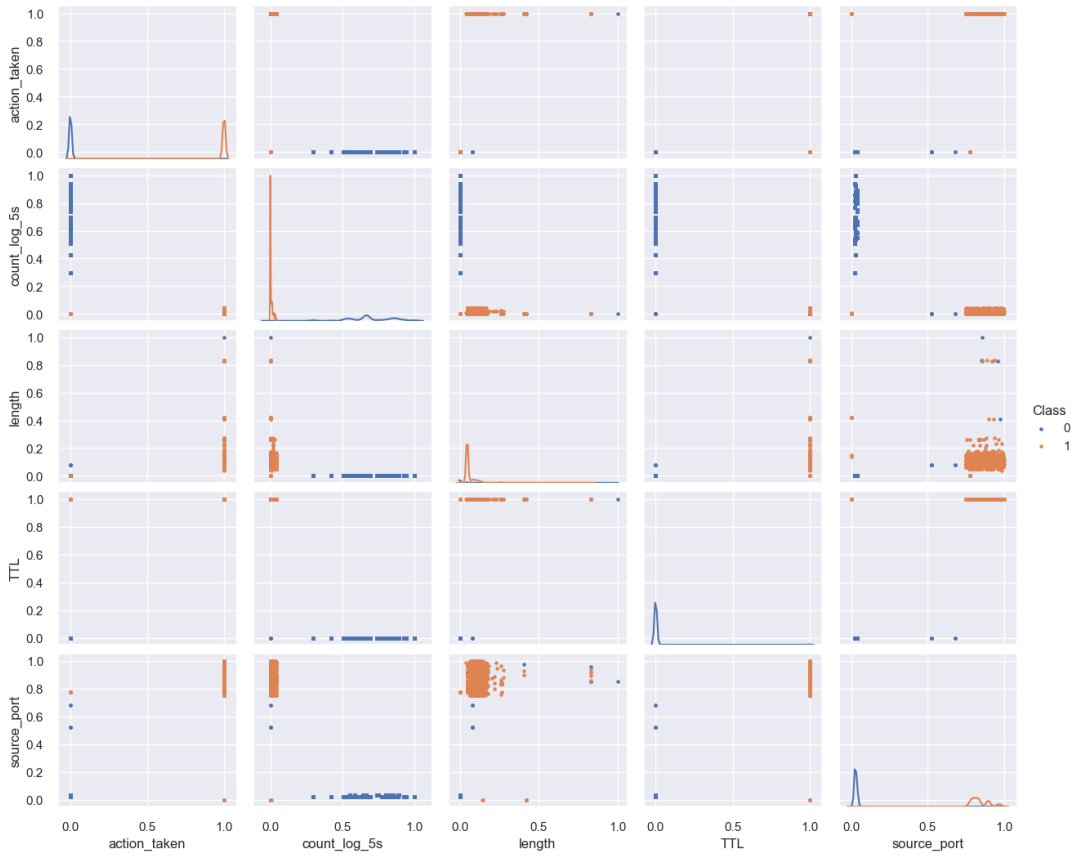

FIGURE 9. Cross correlation of the features.

The listed features have a different relative weight for attack classification purposes. The AI engine allows to classify whether the attack is present or not. Therefore, the classification can assume a binary form.

In order to assess the importance of the values of these features and their role to identify the presence of an attack, a statistical analysis has been carried out about the distribution of the values of each feature in the attack and non-attack conditions.

In Fig. 8 (a)-(n) is shown the statistical distributions of some variables in order to evaluate the value distribution of values feature in the attack and non-attack conditions.

From Fig. 8 it is possible to visually evaluate which parameters are the most representative in case of attack or not. For example, Fig. 8 (n) shows that the 'direction_traffic' parameter is not useful for classification since it is distributed in a similar way between attack and non-attack cases (the distribution of data is the same in both classes). The same consideration applies to the 'data_length', 'delta', 'dest_port' and 'id' parameters shown in Fig. 8 (c), Fig. 8 (d), Fig. 8 (e) and Fig. 8 (m), respectively. On the contrary, the distribution of the 'action_taken', 'count_log_5s', 'window', 'TTL', 'source_port', 'interface_vpn' and 'interface_igb1' parameters shown in Fig. 8 (a), Fig. 8 (b), Fig. 8 (f), Fig. 8 (g), Fig. 8 (h), Fig. 8 (i), and Fig. 8 (l) respectively, indicates that these values are fundamental for classification.

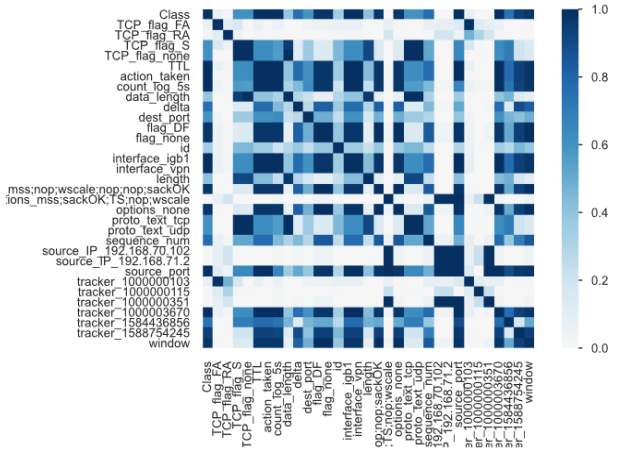

FIGURE 10. Correlation matrix of the features.

To assess in detail the relative distribution of the various classes and any cross-correlation, Fig. 9 shows the distribution of data of five representative variables ('action_taken', 'count_log_5s', 'length', 'TTL', 'source port') in the two classes: attack (Class 1, orange dots) and not attack (Class 0, blue dots). It can be noticed that the data are distributed differently in the two classes, proving that they are useful variables for classification.

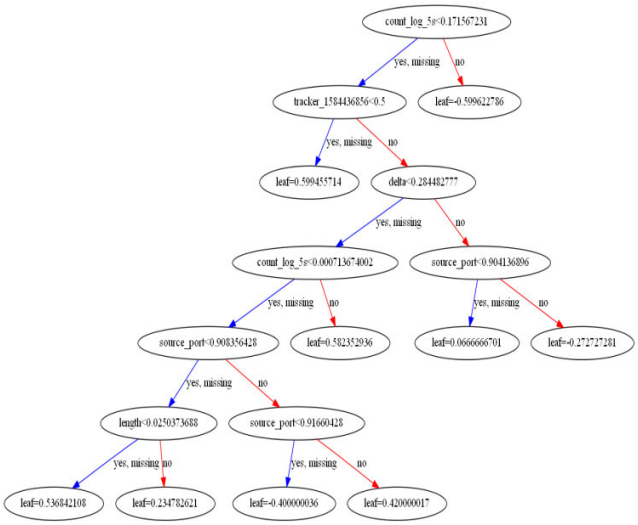

FIGURE 11. XGBoost decisional tree structure.

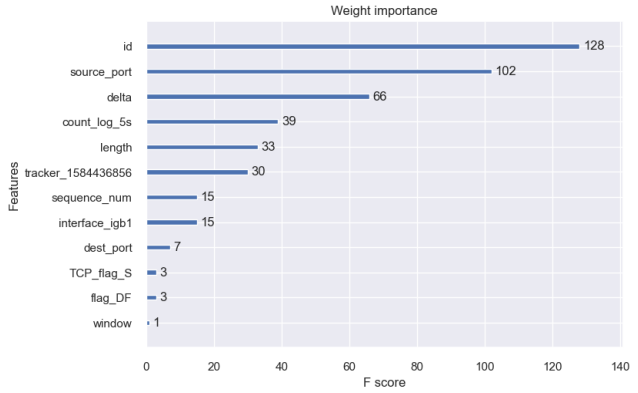

FIGURE 12. Weight importance estimated during the decision tree model construction.

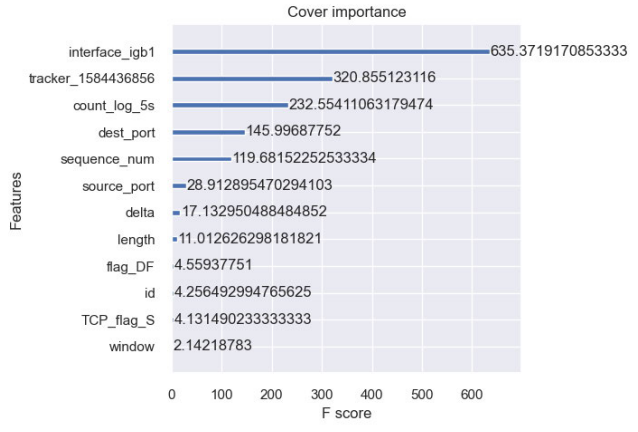

FIGURE 13. Cover importance estimated during the decision tree model construction.

By taking into account all the features, it is possible to calculate the correlation matrix by highlighting similar characteristic trends between variables. Fig. 10 allows to quantify the correlations between the variables used to detect an attack by taking into account all the features in the log files and the information about the related class (attack and non-attack). The correlation matrix is fundamental for classification, since the algorithm must take into account the cross correlations between all the input features. Blue (white) rectangles in Fig. 10 show high (low) correlation between the corresponding features.

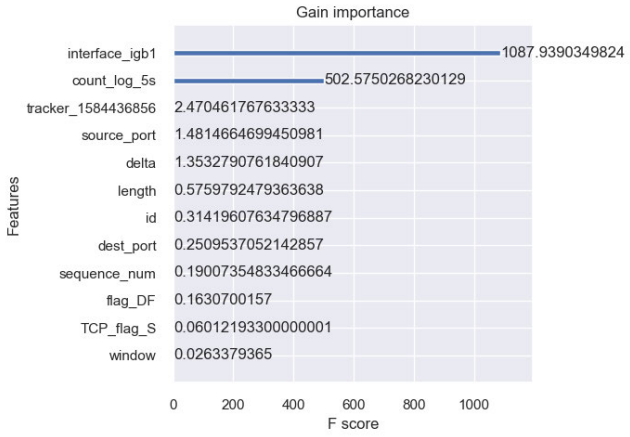

FIGURE 14. Gain importance estimated during the decision tree model construction.

IV. ARTIFICIAL INTELLIGENCE MODEL

In order to select the best AI algorithm to adopt for attack detection and classification, different algorithms have been tested. We select 6 state-of-the-art, commonly used machine learning algorithms for classification purposes. Each model has been created and tested starting from the comparison between the data provided by the model and the real data. The classification performance results are shown in Table 3.

TABLE 3. Performance of the classificator model.

Algorithm	MSE	Accuracy	AUC
XGBoost	0.001	99.9	0.999
Random Forest	0.0012	99.88	0.999
Convolutional Neural Network	0.0102	99.65	0.998
Logistic Regression	0.0033	99.66	0.997
Nearest Neighbors	0.0047	99.53	0.997
Support Vector Machine	0.0034	99.65	0.997

Table 3 shows that all the tested algorithms reach a very high accuracy, i.e. around 99.7%, and an average quadratic error, Mean Squared Error (MSE) of the order of 10^{-3} – 10^{-2} and an AUC score (area under the receiver operating characteristic curve) very close to the unit value. The XGBoost (eXtreme Gradient Boosting) algorithm based on decision trees appears to be the most accurate among those tested. Importantly, our results show that tree-based algorithms (XGBoost and Random Forest) are the most accurate algorithms. This result is consistent with the common belief and recent studies which show that XGBoost algorithms outperform neural network algorithms in handling

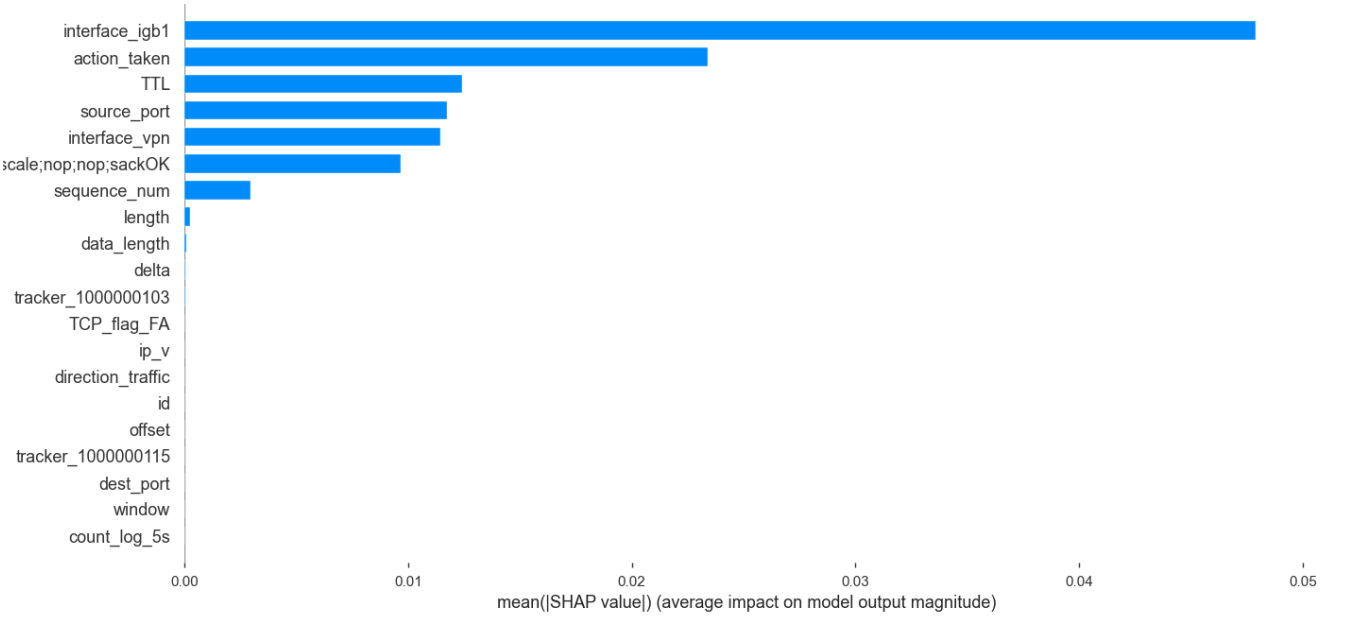

FIGURE 15. Importance of the features in the construction of the decision tree calculated using the SHAP library.

heterogeneous and independent feature in a dataset for classification purposes [65], [66].

The decision tree structure is shown in Fig. 11. The structure of the tree might suggest that the most important variable could be identified as the one related to the initial split of the three, i.e. 'count_log_5s'. Generally, the most representative feature can be identified as the one having the highest weight and influence on the classification algorithm, since it indicates the variable that allows to efficiently build up the tree, thus making the classification process faster by adopting a path across the tree with the fewest number of steps.

Importantly, the use of tree-based algorithm allows to evaluate the importance of the features in the classification process along the branches of the decision tree. The importance of each feature has been computed with different metrics:

- Weight: the number of times a feature is used to move from one branch to another in the decision tree;
- Cover: the number of times a feature is used to move from one branch to another in the decision tree weighed by the number of training data that cross the bifurcations;
- Gain: the average reduction of the training loss when the feature considered is used for creating branches.

These three metrics can be adopted to evaluate the importance of the features in determining whether a cyberattack is ongoing or not. The plots of the importance of the different features using the three metrics are shown in Fig. 12, Fig. 13 and Fig. 14.

The plots show different results, due to the different adopted metric to compute feature importance. This result appears unexpected at a first glance since the different metrics does not univocally identify the most important feature in the classification process. A better way to evaluate the importance of the features is provided by the SHAP Python library [64]. This library allows to estimate the most important feature in a consistent way. According to this metric, the most important feature is defined as that feature whose importance never decreases if we change the data model by modifying the dependency between variables. Figure 15 shows the result of the SHAP algorithm to infer the feature importance. It can be noticed that the 'interface_igb1' plays a fundamental role in the construction of the decision tree, thus representing the most important feature for identifying and classifying cyberattacks.

As regards the Convolutional Neural Network (CNN) model, several network architectures have been built in order to establish the best combination of layers that would minimize the loss function, i.e. the mean squared error. Four convolutive network architectures have been built by adopting different types of (hidden) layers, such as Dropout and Pooling. Figure 16 shows the accuracy and loss function for the four different CNN architecture with increasing training epochs.

We observe that the accuracy and loss functions reach very promising values, comparable with the values reached by other adopted algorithms (see Table 3).

In addition, tests were carried out by training the model with a modified data set by removing the source-IP and tracking feature. The tests show that the accuracy achieved by the model is sufficient and can be used for the classification of cyberattacks. This is due to the fact that the removed variables are among the least representative for the classification (see Fig. 11 - 15) and consequently have less influence on the classification process.

Once the best AI algorithm has been selected, the creation of the model consisted of the following phases:

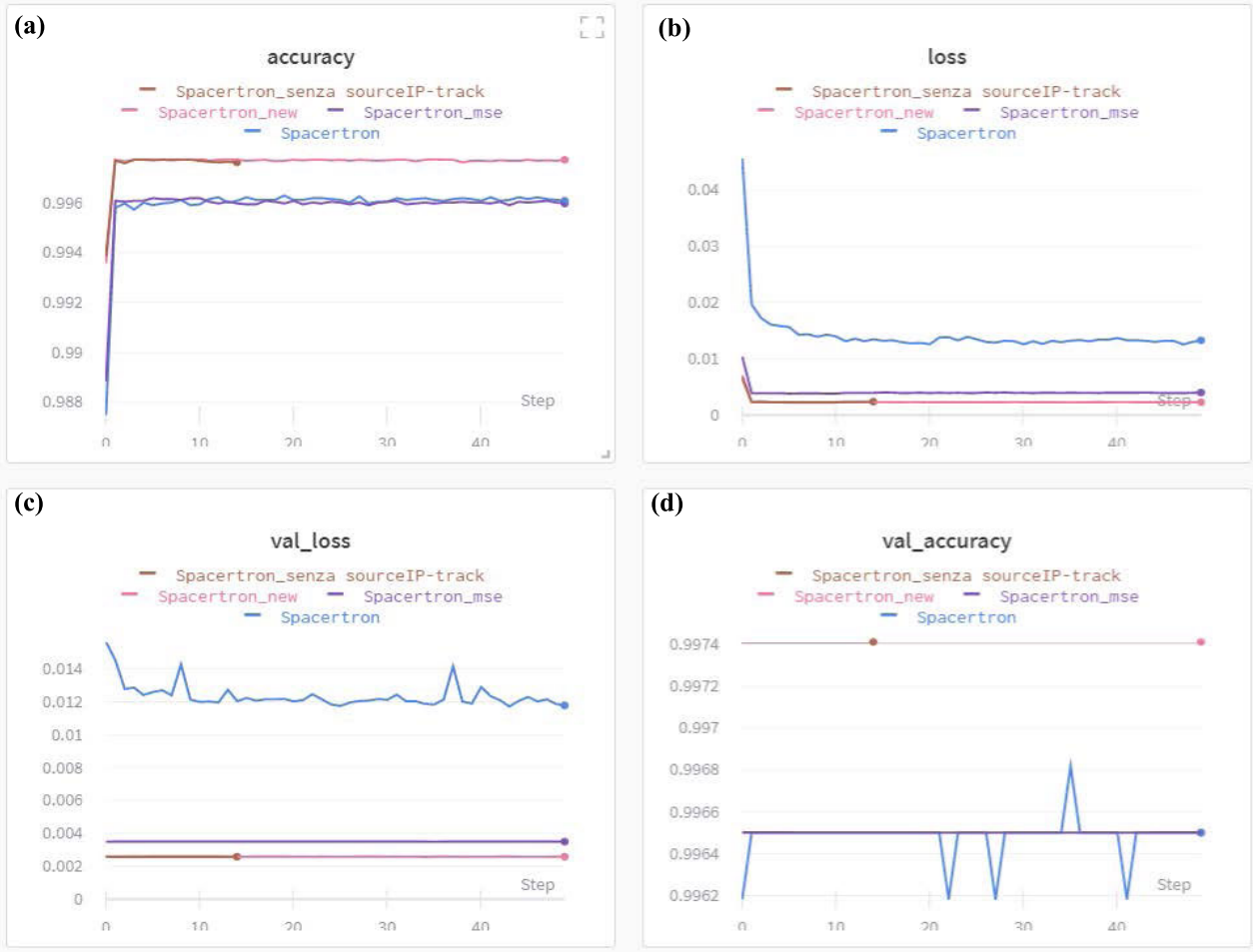

FIGURE 16. Accuracy and loss function varying epochs for the training and validation test of the CNN network.

- pre-processing phase aimed at obtaining a homogeneous form for all the data;
- model creation phase, consisting of application of the XGBoost algorithm for the classification and subsequent validation of the pre-processed data.

The output of the classifier is used to switch the VPN communication on a different available node. Moreover, as previously discussed, the pre-processing activity allowed to evaluate the cross correlation between variables and, through clustering techniques by figuring out the association rules between variables useful for improving the accuracy of the classifier. Furthermore, the pre-classification phase of the attacks and, more generally, the data pre-processing are able to identify association rules on the basis of the temporal recurrence of the attacks. These rules are used for the activation of random switching modes in order to guarantee additional security to the system, after verifying the availability of the nodes. The conditions for random switching are:

- high number of attacks in a limited time compared to daily average;
- high number of connection requests compared to daily average;
- request for connection at the same time of an attack occurred in the previous days.

If these conditions occur, the system checks the availability of the nodes by means of a query and, if necessary, switches the VPN communication on that node. In other words, the engine exploits the classification ability of the XGBoost algorithm to switch the VPN communications and, in order to further increase the security of the platform, adopt the previously mentioned rules for random switching among the available nodes. Finally, the intelligence engine implements the 'self-learning' methodology in order to generate an updated classifier and ensure continuous learning from the most recent log data.

V. MODEL CRYPTOGRAPHY AND DATA SECURITY

In order to further protect the platform against cyberattacks, the AI model was encrypted by means of the SHA-512 hashing algorithm. The Rivest–Shamir–Adleman (RSA) algorithm was used to access the VPN, which generates a pair of keys (public, private) managing the authentication service. Additionally, to further increase security, the private key is subjected to Hash SHA-512 hashing algorithm.

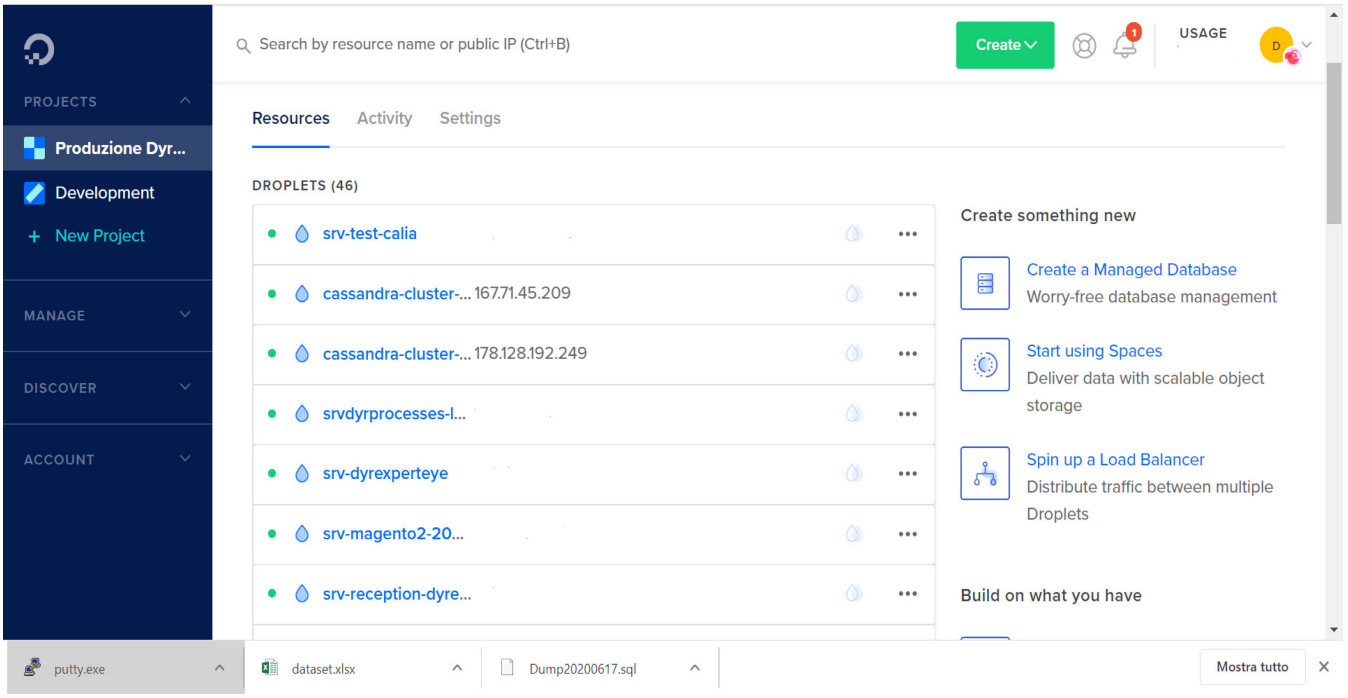

FIGURE 17. Cassandra platform.

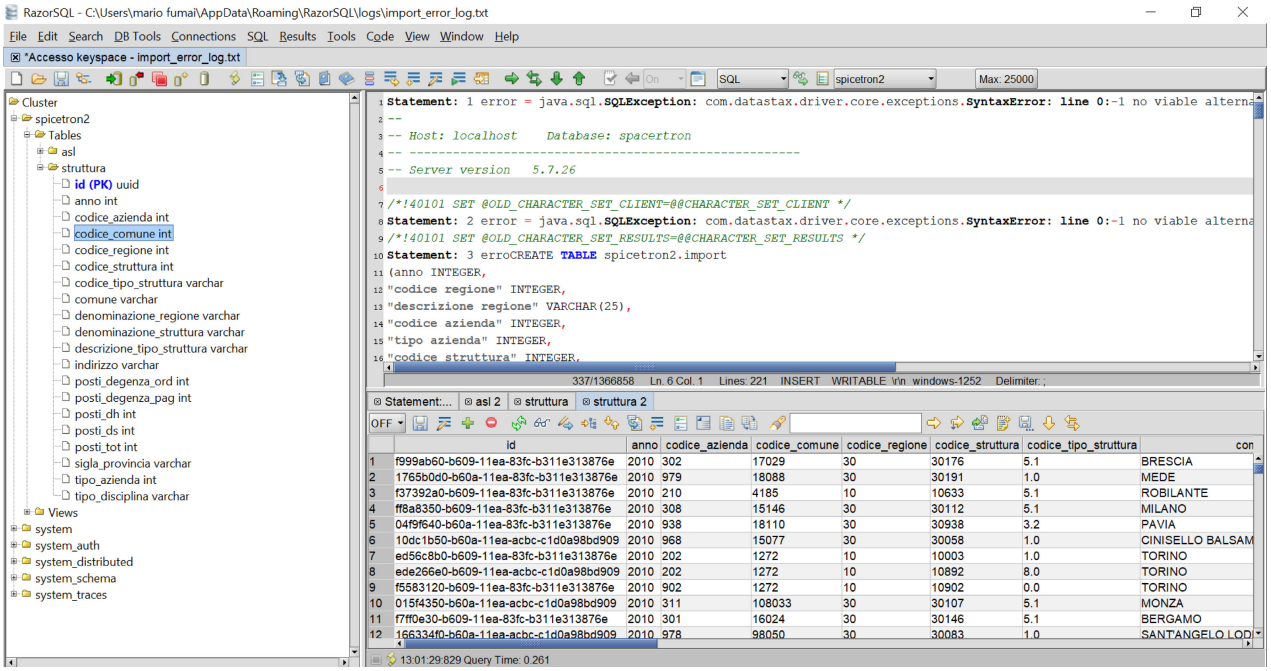

FIGURE 18. RazorSQL platform.

In the event an attack is detected, the control room switches the VPN communications to an available channel that is considered more secure. At the same time, the control room run a script that allows data to be saved and the system to be completely closed to internet traffic. These operations are included in the disaster recovery and data migration systems. In order to manage the storage of the platform, two Big Data Cassandra nodes were configured. Cassandra is one of the most used non-relational database management system distributed with an open source license and it is optimized for the management of large amounts of data. This technology exhibits good computational costs in writing operations [62] compared to other management systems. Specifically, two servers have been configured on the cloud provider of

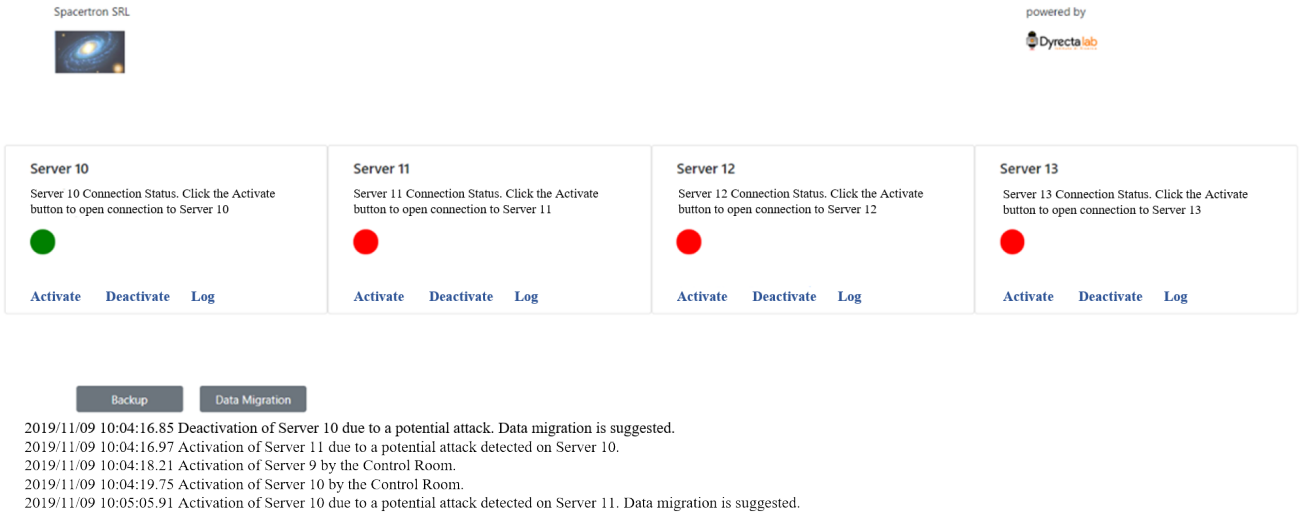

FIGURE 19. Dashboard showing the status of the VPN channels.

"Digital Ocean". Figure 17 shows the platform dashboard with the configuration of the nodes.

On each individual server, the Ubuntu Server version 18.04 LTS and the Cassandra latest stable release (version 3.11) was installed, configuring the cluster between the two nodes and testing synchronization. The "Keyspace" configuration has been performed on the first node and replicated on the second one. The database management was performed by a compatible Client Desktop on the "Windows" operating system called "RazorSQL" (see Fig. 18).

The characteristics of the configured Cassandra environment is replicated on both nodes. The peculiarity of this Big Data management system is the constant synchronization between the nodes that are continuously mirrored between the nodes, always guaranteeing the constant data availability in the event of a failure of one of the two nodes. Moreover, independent benchmark analyses and testing of various NoSQL platforms have been performed in recent years, identifying Apache Cassandra as the best platform in terms of scalability and the management of big data and production-level workloads [67].

VI. PLATFORM TESTING

The development of the system involved the creation of dashboards for displaying the results of the artificial intelligence algorithms and for testing the prototype system.

These dashboards allow to view the configuration of the VPN channels, information on any switching between the channels and the disaster recovery and data migration operations.

The dashboard (shown in Fig. 19) consists of a single screen showing:

- status of the VPN channels;
- summary of system logs with specification of the activation/deactivation of the VPN channels and activation of the disaster recovery and data migration;
- summary of the performance of the AI algorithm used for traffic log classification.

Through these dashboards it is possible to monitor the status of the connections, the history of the detected attacks and the consequent operations carried out such as intelligent switching between channels, disaster recovery and data migration. In addition, it is possible to view the accuracy of the network model used through the graphs of the loss function (MSE) and the accuracy graphs.

VII. CONCLUSION AND DISCUSSION

The results are related to an industrial project enabling a platform managing an intelligent virtualized network capable of controlling the traffic of data packets and assessing the presence of cyberattacks. Our results show that XGBoost algorithm is the most accurate algorithm in identifying and classifying cyberattacks. The development and testing allowed also to create a platform capable of carrying out disaster recovery and data migration operations in order to avoid data loss, make the system less vulnerable to further attacks and guarantee the availability of data. The risks associated with the security of the platform can be divided into different types related to the different behavior of the classification model implemented in the project:

- data not sufficient for a correct model training or not well distributed or missing: the model is unable to perform an accurate classification of the log data if the statistics about attacks are not significant; this case can occur in cases where the pre-trained model is no longer accurate in classification and, therefore, must be retrained with the most recent data which may, in some cases, be insufficient to guarantee a good accuracy;
- incorrect or incomplete data: if some data are not complete, they cannot be processed by the algorithm thus significant data is lost affecting classification accuracy;

- rare events: occurrence of an event not present in the training period that might generate a false positive or a false negative.

Therefore, the classification of the traffic logs needs to be interpreted considering the risks associated with the statistical significance of the data. In order to overcome this problem, a good approach would be to train the model iteratively with new logs data, thus ensuring an accurate classification performance of new potential type of attacks.

APPENDIX

The machine learning libraries used in this work are imported with the following Python script:

```
import numpy as np
import pandas as pd
import seaborn as sns
import shap
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
from sklearn.metrics import mean_squared_error
from sklearn.metrics import accuracy_score
from sklearn.metrics import log_loss
```

FIGURE 20. Python code snippet containing the imported libraries.

```
def ohe(train, categories):
    all_data = train
    for column in
all_data.select_dtypes(include=[np.object]).columns:
        cat = all_data[column].unique()
        train[column] =
pd.Categorical(train[column].astype('category'),
categories = cat)

    for cat in categories:
        dfDum = pd.get_dummies(all_data[cat], prefix=cat)
        all_data =
pd.concat([all_data,dfDum.reindex(sorted(dfDum.columns
),axis=1)], axis=1)
        all_data = all_data.drop(cat, axis=1)
    train_idx = check_col(train)
    train = all_data[all_data.index.isin(train_idx)]
    return train

def standardScale(train, categories):
    scaler = StandardScaler()
    train[categories] =
scaler.fit_transform(train[categories])
    return train
```

FIGURE 21. Python code snippet containing methods for the standardization procedure: ohe (*OneHotEncoding*) and standardScale (*StandardScaler*).

Both the categorical and the quantitative features, during the classifier preprocessing phase, were organized so as to have all the features on the same scale. The quantitative features have been standardized through the *StandardScaler* module implemented in the Python *scikit-learn* library. On the other hand, the One Hot Encoding (OHE) technique (also present in the *scikit-learn* library) was used to standardize the categorical features. The *One Hot Encoding* technique allows to transform each categorical feature containing n possible categories within it into n binary features, each of which representing a category. Standardization and One Hot Encoding is achieved by the methods shown in Fig. 21.

In this way it is possible to standardize all the quantities in order to allow more effective training of the artificial intelligence algorithm.

ACKNOWLEDGMENT

The work has been developed in the framework of the research project: "Digital Transformation of a Prototype Cross Platform oriented on Cybersecurity, Virtual Connectivity, Big Data and Artificial Intelligence Control: "Digital Cross Platform: Virtual AI Distribution Network Processes." Authors gratefully thank the researcher Mario Fumai for the configuration of the Cassandra system.

REFERENCES

[1] *The Measurement of Scientific, Technological and Innovation Activities-Guidelines for Collecting and Reporting Data on Research and Experimental Development*, Frascati, Italy.
[2] M. Rossberg and G. Schaefer, "A survey on automatic configuration of virtual private networks," *Comput. Netw.*, vol. 55, no. 8, pp. 1684–1699, Jun. 2011, doi: 10.1016/j.comnet.2011.01.003.
[3] A. Galán-Jiménez and J. Gazo-Cervero, "Overlay networks: Overview, applications and challenges," *Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 12, pp. 40–49, 2010.
[4] K. Han, J. Liu, D. Yang, and Q. Yuan, "The design of secure embedded VPN gateway," in *Proc. IEEE Workshop Adv. Res. Technol. Ind. Appl. (WARTIA)*, Sep. 2014, pp. 350–353, doi: 10.1109/WARTIA.2014.6976267.
[5] Y. Q. Fan, L. Lv, M. L. Liu, and F. Xie, "Improvements based on the IPSec VPN secuirity," *Adv. Mater. Res.*, vols. 756–759, pp. 2693–2697, Sep. 2013, doi: 10.4028/www.scientific.net/AMR.756-759.2693.
[6] T. Yu and S. Jajodia, Eds., *Secure Data Management in Decentralized Systems*, vol. 33. Boston, MA, USA: Springer, 2007.
[7] A. Ali and M. M. Afzal, "Database Security: Threats and Solutions," *Int. J. Eng. Invent.*, vol. 6, no. 2, pp. 25–27, 2017. [Online]. Available: http://www.ijeijournal.com/papers/Vol.6-Iss.2/D06022527.pdf
[8] J. O. Chan, "An architecture for big data analytics," *Commun. IIMA*, vol. 13, no. 2, p. 1, 2013.
[9] M. Malik and T. Patel, "Database security–attacks and control methods," *Int. J. Inf. Sci. Techn.*, vol. 6, nos. 1–2, pp. 175–183, Mar. 2016, doi: 10.5121/ijist.2016.6218.
[10] H. Wirkuttis and N. Klein, "Artificial intelligence in cybersecurity," *Cyber., Intell. Secur.*, vol. 1, no. 1, pp. 103–119, 2017.
[11] H. M. Rajan, S. Dharani, and V. Sagar, "Artificial intelligence in cyber security—An investigation," *Int. Res. J. Comput. Sci.*, vol. 4, no. 9, pp. 28–30, 2017.
[12] K. Demertzis and L. Iliadis, "A bio-inspired hybrid artificial intelligence framework for cyber security," in *Computation, Cryptography, and Network Security*. Cham, Switzerland: Springer, 2015, pp. 161–193.
[13] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *Proc. 10th Int. Conf. Cyber Conflict (CyCon)*, May 2018, pp. 371–390, doi: 10.23919/CYCON.2018.8405026.

[14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016, doi: 10.1109/COMST.2015.2494502.

[15] J. Kim, J. Kim, H. L. Thi Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2016, pp. 1–5, doi: 10.1109/PlatCon.2016.7456805.

[16] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of recurrent neural networks for botnet detection behavior," in *Proc. IEEE Biennial Congr. Argentina (ARGENCON)*, Jun. 2016, pp. 1–6, doi: 10.1109/ARGENCON.2016.7585247.

[17] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3422–3426, doi: 10.1109/ICASSP.2013.6638293.

[18] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–5, doi: 10.1109/NTMS.2018.8328749.

[19] G. D. Hill and X. J. A. Bellekens, "Deep learning based cryptographic primitive classification," 2017, *arXiv:1709.08385*. [Online]. Available: http://arxiv.org/abs/1709.08385

[20] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 1916–1920, doi: 10.1109/ICASSP.2015.7178304.

[21] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2015, pp. 339–344, doi: 10.1109/NAECON.2015.7443094.

[22] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," Univ. Toledo, Toledo, OH, USA, Tech. Rep. sesa 16(9):e2, 2016, doi: 10.4108/eai.3-12-2015.2262516.

[23] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Secur. Appl.*, vol. 9, no. 5, pp. 205–216, May 2015, doi: 10.14257/ijsia.2015.9.5.21.

[24] X. Hardy, W. Chen, L. Hou, S. Ye, and Y. Li, "DL4MD: A deep learning framework for intelligent malware detection," in *Proc. Int. Conf. Data Mining (DMIN)*, 2016, pp. 61–67.

[25] G. Tzortzis and A. Likas, "Deep belief networks for spam filtering," in *Proc. 19th IEEE Int. Conf. Tools with Artif. Intell. (ICTAI)*, Oct. 2007, pp. 306–309, doi: 10.1109/ICTAI.2007.65.

[26] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," Key Lab. Mach. Perception (MOE), Peking Univ., Beijing, China, Tech. Rep. Corpus ID: 31357403, 2015, pp. 3–15.

[27] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 797–801, doi: 10.1109/ICCNC.2014.6785439.

[28] S. Ranjan, "Machine learning based botnet detection using real-time extracted traffic features," Narus, Inc., Sunnyvale, CA, USA, Tech. Rep. 8682812, 2014.

[29] B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li, "PeerRush: Mining for unwanted P2P traffic," *J. Inf. Secur. Appl.*, vol. 19, no. 3, pp. 194–208, Jul. 2014, doi: 10.1016/j.jisa.2014.03.002.

[30] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. U. R. Ma'arof, and S. Shamshirband, "A study of machine learning classifiers for anomaly-based mobile botnet detection," *Malaysian J. Comput. Sci.*, vol. 26, no. 4, pp. 251–265, Dec. 2013.

[31] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," presented at the 21st USENIX Secur. Symp. (USENIX Security), 2012, pp. 491–506. [Online]. Available: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/antonakakis

[32] T. Chakraborty, F. Pierazzi, and V. S. Subrahmanian, "EC2: Ensemble clustering and classification for predicting Android malware families," *IEEE Trans. Depend. Sec. Comput.*, vol. 17, no. 2, pp. 262–277, Mar. 2020, doi: 10.1109/TDSC.2017.2739145.

[33] C. Annachhatre, T. H. Austin, and M. Stamp, "Hidden Markov models for malware classification," *J. Comput. Virol. Hacking Techn.*, vol. 11, no. 2, pp. 59–73, May 2015, doi: 10.1007/s11416-014-0215-x.

[34] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo, "On the feasibility of online malware detection with performance counters," *ACM SIGARCH Comput. Archit. News*, vol. 41, no. 3, pp. 559–570, Jun. 2013, doi: 10.1145/2508148.2485970.

[35] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phishing Workshop groups 2nd Annu. eCrime Res. Summit (eCrime)*, 2007, pp. 60–69, doi: 10.1145/1299015.1299021.

[36] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, pp. 1–28, Sep. 2011, doi: 10.1145/2019599.2019606.

[37] G. Apruzzese, M. Marchetti, M. Colajanni, G. G. Zoccoli, and A. Guido, "Identifying malicious hosts involved in periodic communications," in *Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2017, pp. 1–8, doi: 10.1109/NCA.2017.8171326.

[38] C. Chen, S. Mabu, K. Shimada, and K. Hirasawa, "Network intrusion detection using class association rule mining based on genetic network programming," *IEEJ Trans. Electr. Electron. Eng.*, vol. 5, no. 5, pp. 553–559, Sep. 2010, doi: 10.1002/tee.20572.

[39] F. Bisio, S. Saeli, P. Lombardo, D. Bernardi, A. Perotti, and D. Massa, "Real-time behavioral DGA detection through machine learning," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2017, pp. 1–6, doi: 10.1109/CCST.2017.8167790.

[40] Y. Ye, D. Wang, T. Li, D. Ye, and Q. Jiang, "An intelligent PE-malware detection system based on association mining," *J. Comput. Virology*, vol. 4, no. 4, pp. 323–334, Nov. 2008, doi: 10.1007/s11416-008-0082-4.

[41] W.-F. Hsiao and T.-M. Chang, "An incremental cluster-based approach to spam filtering," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1599–1608, Apr. 2008, doi: 10.1016/j.eswa.2007.01.018.

[42] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5948–5959, Oct. 2014, doi: 10.1016/j.eswa.2014.03.019.

[43] N. Stanley, R. Kwitt, M. Niethammer, and P. J. Mucha, "Compressing networks with super nodes," *Sci. Rep.*, vol. 8, no. 1, p. 10892, Dec. 2018, doi: 10.1038/s41598-018-29174-3.

[44] J. Crampton, "Practical and efficient cryptographic enforcement of interval-based access control policies," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–30, May 2011, doi: 10.1145/1952982.1952996.

[45] H. Yang, M. Liu, B. Li, and Z. Dong, "A P2P network framework for interactive streaming media," in *Proc. 11th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, vol. 2, 2019, pp. 288–292, doi: 10.1109/IHMSC.2019.10162.

[46] P. Helman, G. Liepins, and W. Richards, "Foundations of intrusion detection (computer security)," in *Proc. Comput. Secur. Found. Workshop V*, 1992, pp. 114–120, doi: 10.1109/CSFW.1992.236783.

[47] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994, doi: 10.1109/65.283931.

[48] Y. Liu, S. Liu, and X. Zhao, "Intrusion detection algorithm based on convolutional neural network," in *Proc. DEStech Trans. Eng. Technol. Res.*, Mar. 2018, doi: 10.12783/dtetr/iceta2017/19916.

[49] W.-H. Lin, H.-C. Lin, P. Wang, B.-H. Wu, and J.-Y. Tsai, "Using convolutional neural networks to network intrusion detection for cyber threats," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Apr. 2018, pp. 1107–1110, doi: 10.1109/ICASI.2018.8394474.

[50] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against Denial-of-Service attacks," *Electronics*, vol. 9, no. 6, p. 916, Jun. 2020, doi: 10.3390/electronics9060916.

[51] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1595–1598, doi: 10.1109/ICDCS.2018.00178.

[52] T. Kim, S. C. Suh, H. Kim, J. Kim, and J. Kim, "An encoding technique for CNN-based network anomaly detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2960–2965, doi: 10.1109/BigData.2018.8622568.

[53] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against Denial-of-Service attacks," *Electronics*, vol. 9, no. 6, p. 916, Jun. 2020, doi: 10.3390/electronics9060916.

[54] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[55] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.
[56] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.
[57] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.
[58] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
[59] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.
[60] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, "Network intrusion detection: Based on deep hierarchical network and original flow data," *IEEE Access*, vol. 7, pp. 37004–37016, 2019, doi: 10.1109/ACCESS.2019.2905041.
[61] A. Massaro, G. Panarosa, N. Savino, S. Buonopane, and A. Galiano, "Advanced multimedia platform based on big data and artificial intelligence improving cybersecurity," *Int. J. Netw. Secur. Appl.*, vol. 12, no. 3, pp. 23–37, May 2020, doi: 10.5121/ijnsa.2020.12302.
[62] M. D'Aloia, R. Russo, G. Cice, A. Montingelli, G. Frulli, E. Frulli, F. Mancini, M. Rizzi, and A. Longo, "Big data performance and comparison with different DB Systems," *Int. J. Comp. Sci. Inform. Technol.*, vol. 8, no. 1, pp. 59–63, 2017.
[63] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, "Cybersecurity for industrial control systems: A survey," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101677.
[64] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nature Mach. Intell.*, vol. 2, no. 1, pp. 56–67, Jan. 2020, doi: 10.1038/s42256-019-0138-9.
[65] R. S. Olson, W. La Cava, Z. Mustahsan, A. Varik, and J. H. Moore, "Data-driven advice for applying machine learning to bioinformatics problems," 2017, *arXiv:1708.05070*. [Online]. Available: http://arxiv.org/abs/1708.05070
[66] N. Memon, S. B. Patel, and D. P. Patel, "Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, 2019, pp. 452–460, doi: 10.1007/978-3-030-34869-4_49.
[67] T. Rabl, S. Gómez-Villamor, M. Sadoghi, V. Muntés-Mulero, H.-A. Jacobsen, and S. Mankovskii, "Solving big data challenges for enterprise application performance management," *Proc. VLDB Endowment*, vol. 5, no. 12, pp. 1724–1735, Aug. 2012, doi: 10.14778/2367502.2367512.

ALESSANDRO MASSARO (Senior Member, IEEE) (Professor, ING/INF/01, FIS/01, FIS/03) carried out scientific research at the Polytechnic University of Marche, at CNR, and at the Italian Institute of Technology (IIT) as the Team Leader by activating laboratories for nanocomposite sensors for industrial robotics. He is in MIUR register as Scientific Expert in competitive industrial research and social development, and he is currently the Head of the Research and Development Section and the Scientific Director of MIUR Research Institute, Dyrecta Lab S.r.l., and a member of the International Scientific Committee of Measurers IMEKO. He recently received an award from the National Council of Engineers as the Best Engineer of Italy 2018 (Top Young Engineer 2018). Actually, he is an Associate Editor of IEEE Access.

MICHELE GARGARO received the master's degree in information technology and cybersecurity from the University of Bari, Italy, in 2019. He is currently a Researcher with the Dyrecta Laboratory Institute studying topics about artificial intelligence applied on cybersecurity in industrial applications.

GIOVANNI DIPIERRO received the master's degree in physics and the Ph.D. degree in physics, astrophysics, and applied physics from the University of Milan, Italy, in 2014 and 2017, respectively After receiving his Ph.D. degree, he has worked as a Postdoctoral Researcher with the University of Leicester, U.K. In 2019, he joined the Research Team, Dyrecta Laboratory Srl, Conversano, Italy. His research interest includes developing artificial intelligence algorithms to analyze data and performing numerical simulations of fluid dynamics and structural mechanics over a large range of physical phenomena.

ANGELO MAURIZIO GALIANO received the M.S. degree in education science in 2009. He is currently the CEO of Dyrecta Laboratory Srl–Research Institute accredited by the Italian Ministry of University and Scientific Research. He has more than 20 years of experience in the field of information technologies. His current research interests include neural networks, smart health, and predictive analytics.

SIMONE BUONOPANE is currently the CEO and the Founder of Spacertron (a high value technology company). He has been involved in Enterprise, Government, and Cybersecurity Hw/Sw applications/solutions in many different countries with a specific focus on innovation, research, and high availability.